**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

Курсовая РАБОТА

**по дисциплине «Программирование»**

Тема: Хранение и обработка числовых данных на основе файлов и массивов

|  |  |  |
| --- | --- | --- |
| Студент гр. 4354 |  | Чучалин И. В. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Чучалин И.В. | | |
| Группа 4354 | | |
| Тема работы: хранение и обработка числовых данных на основе файлов и массивов. | | |
| Исходные данные: реализовать алгоритм на основе разбиения программы на набор функций с параметрами. Программа не должна содержать стандартных и библиотечных типов и связанных с ними функций. Должны быть учтены все составляющие формулировки задания и выполнены указанные действия, причем в качестве результата должны быть предложены все варианты, удовлетворяющие заданию. Программа должна предоставлять возможность неоднократного выполнения действий. | | |
| Содержание пояснительной записки:  Исходная формулировка задания, математическая постановка задачи, описание алгоритма, текст программы, вывод программы, выводы. | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания: 22.11.2024 | | |
| Дата сдачи реферата: | | |
| Дата защиты реферата: | | |
| Студент |  | Чучалин И.В. |
| Преподаватель |  | Калмычков В.А. |

**Аннотация**

В данной курсовой работе рассмотрена работа с числовыми данными, файлами, массивами и функциями на примере языка C++. Необходимо корректно составить математическую постановку задачи, обработать файл с исходными данными, при помощи двухмерных или трёхмерных массивов организовать хранение данных и с помощью функций реализовать программу для решения геометрической задачи. Результатом выполнения курсовой работы служит программа, способная решать геометрическую задачу с разными начальными условиями и реагирующая на некорректные данные.

**Summary**

In this course work, work with numeric data, files, arrays and functions is considered using the example of the C++ language. It is necessary to correctly formulate a mathematical statement of the problem, process the file with the initial data, organize data storage using two-dimensional or three-dimensional arrays and use functions to implement a program to solve a geometric problem. The result of the course work is a program capable of solving a geometric problem with different initial conditions and responding to incorrect data.

Содержание:

[Исходная формулировка задания 4](#_Toc184169335)

[Ограничения на исходные данные 5](#_Toc184169336)

[Математическая постановка задачи 5](#_Toc184169337)

[Дано 5](#_Toc184169338)

[Найти 5](#_Toc184169339)

[Способ решения. 5](#_Toc184169340)

[Контрольный пример 6](#_Toc184169341)

[Организация интерфейса пользователя 6](#_Toc184169342)

[Макет ввода/вывода 6](#_Toc184169343)

[Реализация вывода в консоль 7](#_Toc184169344)

[Внутренний формат хранения данных 7](#_Toc184169345)

[Работа с файлами 8](#_Toc184169346)

[Описание функций 8](#_Toc184169347)

[Синтаксис 8](#_Toc184169348)

[Назначение 9](#_Toc184169349)

[Описание алгоритма 10](#_Toc184169350)

[Текст программы 15](#_Toc184169351)

[Вывод программы 19](#_Toc184169352)

[Выводы 20](#_Toc184169353)

# Исходная формулировка задания

Дано N точек на плоскости. Для всех треугольников, образуемых любыми тремя точками, определить, является ли треугольник остроугольным.

# Ограничения на исходные данные

Будем обрабатывать не больше 100 точек ().

# Математическая постановка задачи

Дано точек на плоскости.

Найтивсе остроугольные треугольники, образуемые любыми тремя точками.

Способ решения.

Контрольный пример

# Организация интерфейса пользователя

## Макет ввода/вывода

|  |  |
| --- | --- |
| Fresult1. | (d\_\_d.d\_\_d;d\_\_d.d\_\_d), (d\_\_d.d\_\_d;d\_\_d.d\_\_d), (d\_\_d.d\_\_d;d\_\_d.d\_\_d), (d\_\_d.d\_\_d;d\_\_d.d\_\_d) |
| Fresult2. | ЗАДАНИЕ:  Дано N точек на плоскости. Найти все пары квадратов с вершинами в этих точках, имеющих наибольшую площадь пересечения. Вывод вершин осуществить против часовой стрелки. Стороны квадратов расположены параллельно.  ОБРАБОТКА ДАННЫХ: |
| Fresult3. | Кол-во точек = 0. |
| Fresult4. | На обработку взято d\_\_d точек. Построено d\_\_d квадратов. |
| Fresult5. | РЕЗУЛЬТАТ: |
| Fresult6. | Наибольшая площадь пересечения, равная = d\_\_d, у |
| Fresult7. | квадрата с вершинами в точках |
| Fresult8. | и квадрата с вершинами |
| Fresult9. | Никакие квадраты не пересекаются. |
| Fresult10. | Файл пуст. |
| Fresult11. | Кол-во точек не может быть отрицательным. |
| Fresult12. | Кол-во точек = 0. |
| Fresult13. | Слишком большое кол-во точек. |
| Fresult14. | Не удалось открыть протокол. |
| Fresult15. | Не удалось открыть файл с входными данными. |
| Fprotocol1. | Пустая строка. |
| Fprotocol2. | Не удалось взять точку (d\_\_d.d\_\_d;?). Недостаточно данных. |
| Fprotocol3. | Получена d\_\_d.d\_\_d -я точка с координатами (d\_\_d.d\_\_d;d\_\_d.d\_\_d) |
| Fprotocol4. | Точка (d\_\_d.d\_\_d;d\_\_d.d\_\_d) уже записана. |
| Fprotocol5. | Построен d\_\_d-й квадрат с вершинами |
| Fprotocol6. | Не удалось построить d\_\_d-й квадрат с вершинами |
| Fprotocol7. | Площадь пересечения d\_\_d-го и d\_\_d-го квадратов равна = |
| Fprotocol8. | d\_\_d-й и d\_\_d-й квадраты не пересекаются. |
| Cout1. | Не удалось открыть файл для вывода результата. |
| Fpoints1. | s\_s |
| Fpoints2. | d\_\_d.d\_\_d |

## Реализация вывода в консоль

|  |  |
| --- | --- |
| Библиотека | Вывод |
| iostream | std::cout << |

## Внутренний формат хранения данных

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Назначение |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Работа с файлами

Для работы с файлами используется библиотека «fstream».

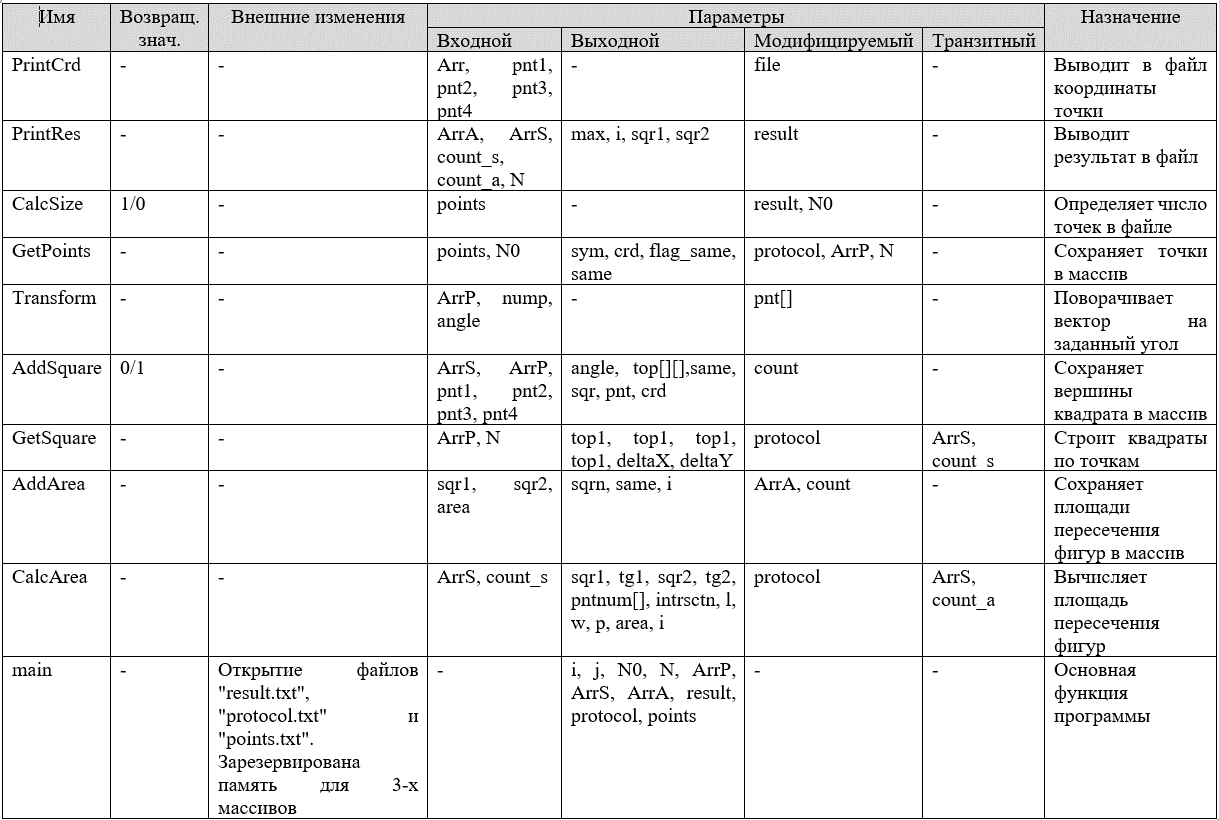
|  |  |
| --- | --- |
| Метод | Описание |
| fstream f | Создание файловой переменной |
| f.open(“file.txt”, ios::out) | Открытие файла на запись |
| f.open(“file.txt”, ios::in) | Открытие файла на чтение |
| filein.seekg(-1, ios::cur); | Перемещение курсора назад относительно текущего положения |
| filein >> noskipws | Чтение пробелов, символов перехода на новую строку |
| f.is\_open() | Определяет, открыт ли файл |
| f.eof() | Проверка на конец файла |
| f << | Запись данных в файл |
| f >> | Чтение данных из файла |

# Описание функций

## Синтаксис

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип возвращаемого значения | Параметры | | |
| Тип | Имя | Изменение |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| main | void | - | - | - |

## Назначение

****

# Описание алгоритма

Программа Результат работы программы выводится в файл «result.txt». Все действия программы фиксируются в протоколе «protocol.txt». Блок-схема работы программы представлена ниже:

# Текст программы

# Вывод программы

Контрольный пример: входной файл (рис. 5), файл с результатом (рис. 6), протокол (рис. 7).

# Выводы

В ходе данной курсовой работы были рассмотрены следующие понятия: обработка и хранение числовых данных, файловые переменные, режимы открытия и закрытия файла, проверка на то, открыт ли файл, считывание и запись информации в файл, динамическая память, указатели, арифметика указателей, удаление данных, одномерные, двумерные и трёхмерные массивы, функции, параметры функций, возвращаемое значение функций, ссылки, типы данных, условные конструкции, циклы и функция main. Также на практике была реализована программа, решающая геометрическую задачу на пересечение фигур на координатной плоскости.